# An Evolutionary Multi-Objective Local Selection Algorithm for Customer Targeting


**YongSeog Kim**
Management Sciences Dept.
University of Iowa
Iowa City, IA 52242 USA
yong-s-kim@uiowa.edu

**W. Nick Street**
Management Sciences Dept.
University of Iowa
Iowa City, IA 52242 USA
nick-street@uiowa.edu

**Filippo Menczer**
Management Sciences Dept.
University of Iowa
Iowa City, IA 52242 USA
filippo-menczer@uiowa.edu



**Abstract- In an increasingly competitive marketplace, one of the most interesting and challenging problems is how to identify and profile customers who are most likely to be interested in new products or services. At the same time, minimizing the number of variables used in the prediction task is important with large databases. In this paper we consider a novel application of evolutionary multi-objective algorithms for customer targeting. Evolutionary algorithms are considered effective in solving multi-objective problems because of their inherent parallelism. We use ELSA, an evolutionary local selection algorithm that maintains a diverse population of solutions approximating the Pareto front in a multi-dimensional objective space. We use artificial neural networks (ANNs) for customer prediction and ELSA to search for promising subsets of features. Our results on a real data set show that our approach is easier to interpret and more accurate than the traditional method used in marketing.**


## 1 Introduction

In the last decade, evolutionary multi-objective (EMO) algorithms have been applied to solve many engineering and scientific problems [20, 10, 16, 3]. In multi-objective optimization problems, we expect a set of optimal solutions rather than a single optimal solution because each solution should be evaluated on multiple objectives and often such objectives are in conflict with each other. In particular, we are interested in finding a non-dominated solution set or a *Pareto* optimal set.[1] We note that many business-related problems can also be modeled using the EMO framework. The multi-objective nature of the decision making process in various business applications makes EMO algorithms ideal to provide human decision makers with a set of *good* candidate solutions, and for exploring the trade-offs among the various objectives.

For instance, most market managers realize that the traditional, one-to-one customer relationships have largely disappeared and that consumer behavior has become infinitely more volatile and difficult to predict. Rapidly changing demographic patterns force more and more organizations to eagerly look for recognizable sub-populations of customers who have similar behavioral patterns and who may be open to targeted marketing messages to maintain their competitive market edge [1]. At the same time, with the size of databases growing rapidly, data dimensionality reduction becomes another important factor in building a prediction model that is fast, easy to interpret, cost effective, and generalizes well to unseen cases. Principal Component Analysis (PCA) [6] and logistic regression have been among the most popular models in the marketing industry for data reduction and prediction, respectively. In this study, we propose a new approach that combines EMO algorithms for data reduction and artificial neural networks (ANNs) for the customer prediction task. Our approach is different from previous studies on direct marketing in the sense that they did not consider either multiple objectives [11] or data reduction [1].

Data reduction is performed via feature selection in our approach. Feature selection is defined as the process of choosing a subset of the original predictive variables by eliminating redundant features and those with little or no predictive information. If we extract as much information as possible from a given data set while using the smallest number of features, we can not only save a great amount of computing time and cost, but often build a model that generalizes better to unseen points. Feature selection can also significantly improve the comprehensibility of the resulting classifier models. Even a complicated model — such as a neural network — can be more easily understood if constructed from only a few variables. In marketing applications it can also be useful to communicate the set of good predictive variables to the sales staff, giving them rules of thumb for targeting potential customers. We adopt the wrapper model [9] of feature selection which requires two components: a search algorithm that explores the combinatorial space of feature subsets, and one or more criterion functions that evaluate the quality of each subset based directly on the predictive model.

An evolutionary algorithm is used to search through the possible combinations of features, and two quality measurements, hit rate (which should be maximized) and complexity (which should be minimized), are used to evaluate the quality of each feature subset. Many popular EMO algorithms do not consider such objectives separately but create new single objective in a subjective manner as noted in [2, 21]. We instead use the Evolutionary Local Search Algorithm (ELSA) [13], which maintains a diverse population of solutions approximating the Pareto front in a multi-dimensional objective space. Unlike other evolutionary algorithms and their vari-

[1] According to [3], a Pareto optimal set becomes a non-dominated solution set when the explored sample space is the same as the entire search space.

ants [18, 5, 19, 4], ELSA performs a "local" search in the space of feature subsets by evaluating each individual based on both its quality measurements and on the number of similar individuals in its neighborhood in objective space. ELSA's bias toward diversity makes it ideal for multi-objective optimization, giving the decision maker a clear picture of Pareto-optimal solutions from which to choose. Previous research has demonstrated the effectiveness of ELSA for feature selection in both supervised [14] and unsupervised [8] learning.

The input features selected by the ELSA individual are used to train an ANN that predicts "buy" or "not buy." Through provided examples, an ANN is able to learn typical patterns of customers in the data set. The trained ANN is tested on an evaluation set, and the individual is evaluated both on the hit rate and the complexity (number of features) of the solution. The result is a predictive model that uses only a subset of the original features, thus simplifying the model and reducing the risk of overfitting while maintaining accuracy, shortening analysis time, and lowering the cost of collecting unnecessary features in the future.

In Section 2, we illustrate ELSA in detail. In Section 3, we show the structure of the ELSA/ANN model and review the feature subset selection procedure and evaluation metrics. We also explain the problem and the data set in detail. In Section 4, we present results from experiments with both the ELSA/ANN and PCA/logit model, and compare them in terms of hit rate and complexity.

# 2 Evolutionary Local Selection Algorithm (ELSA)

## 2.1 Local Selection and Algorithm Details

ELSA springs from algorithms originally motivated by artificial life models of adaptive agents in ecological environments [12]. Modeling reproduction in evolving populations of realistic organisms requires that selection, like any other agent process, be locally mediated by the environment in which the agents are situated.

In a standard evolutionary algorithm, an agent is selected for reproduction based on how its fitness compares to that of other agents. In ELSA, an agent (candidate solution) may die, reproduce, or neither based on an endogenous energy level that fluctuates via interactions with the environment. The energy level is compared against a constant selection threshold for reproduction. By relying on such *local* selection, ELSA reduces the communication among agents to a minimum. The competition and consequent selective pressure is driven by the environment [15]. Further, the local selection scheme naturally enforces the diversity of the population, making ELSA appropriate for multi-objective optimization problems. According to a comparative study of EMO algorithms on feature selection problems in [15], ELSA showed superior coverage of objective space for feature subsets compared to NPGA and remained competitive in terms of accuracy. A more extensive discussion of the algorithm and its application to Pareto optimization problems can be found elsewhere [14, 15]. Figure 1 outlines the ELSA algorithm at a high level of abstraction for feature selection problems.

```
initialize population of agents, each with energy θ/2
while there are alive agents and for T iterations
   for each energy source c
      for each v (0 .. 1)
         E^c_envt(v) ← 2v E^c_tot
      endfor
   endfor
   for each agent a
      a' ← mutate(clone(a))
      for each energy source c
         v ← Fitness(a', c)
         ΔE ← min(v, E^c_envt(v))
         E^c_envt(v) ← E^c_envt(v) − ΔE
         E_a ← E_a + ΔE
      endfor
      E_a ← E_a − E_cost
      if (E_a > θ)
         insert a' into population
         E_a' ← E_a/2
         E_a ← E_a − E_a'
      else if (E_a < 0)
         remove a from population
      endif
   endfor
endwhile
```

Figure 1: ELSA pseudo-code. In each iteration, the environment is replenished and then each living agent executes the main loop. In sequential implementations, the main loop calls agents in random order to prevent sampling effects. We stop the algorithm after $T$ iterations.

## 2.2 Agents, Mutation and Selection

Each agent in the population is first initialized with some random solution and an initial reservoir of *energy*. The representation of an agent consists of $D$ bits, with each of the bits indicating whether the corresponding feature is selected or not (1 if a feature is selected, 0 otherwise).

Mutation is the main operator used to explore the search space, and the crossover operator could be added if required depending on the problem domain. The mutation operator randomly selects one bit of each agent and mutates it. At each iteration an agent produces a mutated clone to be evaluated. Each agent competes for a scare resource, energy, based on its multi-dimensional fitness and the proximity of other agents in the solution space. In the selection part of the algorithm, each agent compares its current energy level with a fixed threshold $\theta$. If its energy is higher than $\theta$, the agent reproduces: the mutated clone that was just evaluated becomes part of the population, with half of its parent's energy. When an agent runs out of energy, it is killed.

The population size is maintained dynamically over the iterations and is determined by the carrying capacity of the environment depending on the costs incurred by the agents and on the replenishment of resources, both described below [15]. The population size is also independent of the reproduction threshold, $\theta$, which only affects the energy stored by the population at steady-state.

### 2.3 Energy Allocation and Replenishment

In each iteration of the algorithm, an agent explores a candidate solution (the mutated clone). The agent collects $\Delta E$ from the environment and is taxed with a constant cost $E_{cost}$ ($E_{cost} < \theta$) for this "action." The net energy intake of an agent is determined by its fitness. This is a function of how well the candidate solution performs with respect to the criteria being optimized. But the energy also depends on the state of the environment. The environment corresponds to the set of possible values for each of the criteria being optimized.[2] We imagine an energy source for each criterion, divided into bins corresponding to its values. So, for criterion fitness $F_c$ and bin value $v$, the environment keeps track of the energy $E^c_{envt}(v)$ corresponding to the value $F_c = v$. Further, the environment keeps a count of the number of agents $P_c(v)$ having $F_c = v$. The energy corresponding to an action (alternative solution) $a$ for criterion $F_c$ is given by

$$Fitness(a, c) = \frac{F_c(a)}{P_c(F_c(a))}. \qquad (1)$$

Candidate solutions receive energy only inasmuch as the environment has sufficient resources; if these are depleted, no benefits are available until the environmental resources are replenished. Thus an agent is rewarded with energy for its high fitness values, but also has an interest in finding unpopulated niches in objective space, where more energy is available. The result is a natural bias toward diverse solutions in the population.

When the environment is replenished with energy, each criterion $c$ is allocated an equal share of energy:

$$E^c_{tot} = \frac{p_{max} E_{cost}}{C} \qquad (2)$$

where $C$ is the number of criteria considered. This energy is apportioned in linear proportion to the values of each fitness criterion, so as to bias the population toward more promising areas in objective space. Note that the total replenishment energy that enters the system at each iteration is $p_{max} \cdot E_{cost}$, which is independent of the population size $p$ but proportional to the parameter $p_{max}$. This way we can maintain $p$ below $p_{max}$ on average, because in each iteration the total energy that leaves the system, $p \cdot E_{cost}$, cannot be larger than the replenishment energy.

## 3 ELSA/ANN Model for Customer Targeting

### 3.1 Problem Specification

Direct mailings to potential customers have been one of the most used approaches to market a new product or service. However, most receivers are not interested in this "junk" mail and throw it away as trash. The result is that the company wastes its time and money to collect contact information and to send mail for very little return. Receivers also have suffered "mail pollution" and the community has lost human and natural resources. If the company had a better understanding of who their potential customers were, they would know more accurately whom to target, and they could reduce expenses and the waste of time and effort.

In this study, we are specifically interested in predicting potential customers who would be interested in buying a recreational vehicle (RV) insurance policy[3] while reducing feature dimensionality. Insurance companies usually have good socio-demographic data regarding their business area. This kind of data is easily available even for one who has not yet started a business. Suppose that one insurance company wants to advertise a new insurance policy based on socio-demographic data over a certain geographic area. From its first direct mailing to 5822 prospects, 348 purchased RV insurance, resulting in a hit rate of $348/5822 = 5.97\%$. Although this is not bad, the company hopes for a higher response rate from another carefully chosen direct mailings from the top $x\%$ of a new set of 4000 potential prospects over the same geographic area. How could it attain a higher hit rate?

### 3.2 Data Sets

In our experiment, we use two separate data sets, a training set with 5822 records and an evaluation set with 4000 records. The training data is used to train ANNs and estimate the expected hit rate on the evaluation set. The evaluation data is used to validate the evolved predictive models. A predictive model returns the top $x\%$ of customers who are most likely to buy RV insurance, and the percentage can easily be adjusted based on the costs and returns of the marketing campaign. Based on a list of chosen prospects, the actual hit rate can be calculated from the evaluation set.

Originally, each data set had 85 attributes, containing socio-demographic information (attributes 1-43) and contribution to and ownership of various insurance policies (attributes 44-85). The socio-demographic data was derived using zip codes and thus all customers living in areas with the same zip code have the same socio-demographic attributes. We omitted the first feature (customer subtype) mainly because it would expand search space dramatically with little information gain if we represented it as a 41-bit variable. Further we can still exploit the information of customer type by recording the fifth feature (customer main type) as a 10-bit variable. The other features are considered continuous and scaled to a common range (0–9).

### 3.3 ELSA/ANN Model Specification

#### 3.3.1 Structure of the ELSA/ANN Model

Our predictive model is a hybrid model of the ELSA and ANN procedures, as shown in Figure 2. ELSA searches for a

[2] Continuous objective functions are discretized.

[3] This is one of main tasks in the 2000 CoIL challenge [7]. For more information about CoIL challenges and the data sets, please refer to `http://www.dcs.napier.ac.uk/coil/challenge/`.

set of feature subsets and passes it to an ANN. The ANN extracts predictive information from each subset and learns the patterns using a randomly selected 2/3 of the training data. The learning algorithm is standard back-propagation of error [17]. The trained ANN is then evaluated on the remaining 1/3 of the training data, and returns two evaluation metrics, $F_{accuracy}$ and $F_{complexity}$ (described below), to ELSA.

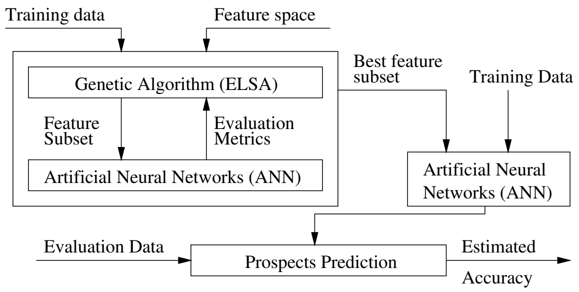


Figure 2: The ELSA/ANN model. ELSA searches for a good subset of features and passes it to an ANN. The ANN estimates the quality of each subset and returns two evaluation metrics to ELSA.

It is important to note that in both the learning and evaluation procedures, the ANN uses only the selected features. Based on the returned metric values, ELSA biases its search direction to maximize the two objectives. This routine continues until the maximum number of iterations is attained. All evaluated solutions over the generations are saved into an offline solution set without comparison to previous solutions. In this way, high-quality solutions are maintained without affecting the evolutionary process.

Among all the evaluated subsets, we choose for further evaluation the set of candidates that satisfy a minimum hit rate threshold. With these chosen candidates, we start a more rigorous selection procedure, 10-fold cross validation. In this procedure, the training data is divided into 10 non-overlapping groups. We train an ANN using the first nine groups of training data and evaluate the trained ANN on the remaining group. We repeat this procedure until each of the 10 groups has been used as a test set once. We take the average of the accuracy measurements over the 10 evaluations and call it an *intermediate* accuracy. We repeat 10-fold cross validation procedure five times and average the five intermediate accuracy estimates. We call this the *estimated* accuracy through the following sections.

We maintain a superset of the Pareto front containing those solutions with the highest accuracy at every $F_{complexity}$ level covered by ELSA. For evaluation purposes, we select a single "best" solution in terms of both estimated accuracy and complexity. We subjectively decided to pick a solution with the minimal number of features at the marginal accuracy level.[4] Once we decide on the best solution, we train the ANN using all the training data with the selected features only. The trained model is then used to rank the potential customers (the records in the evaluation set) in descending order by the probability of buying RV insurance, as predicted by the ANN. We finally select the top $x\%$ of the prospects and calculate the *actual* accuracy of our model using the actual choices of the evaluation set households.

### 3.3.2 Evaluation Metrics

We use two heuristic evaluation criteria, $F_{accuracy}$ and $F_{complexity}$, to evaluate selected feature subsets. Each objective, after being normalized into 25 intervals to allocate energy, is to be maximized by ELSA.

**$F_{accuracy}$:** The purpose of this objective is to favor feature sets with a higher hit rate. Each ANN takes a selected set of features to learn data patterns and predicts which potential customers will actually purchase the product. We define two different measures, $F^1_{accuracy}$ and $F^2_{accuracy}$ for two different experiments. In experiment 1, we select the top 20% of potential customers in descending order of the probability of purchasing the product and compute the ratio of the number of actual customers, $AC$, out of the chosen prospects, $TC$. We calculate $F^1_{accuracy}$ as follows:

$$F^1_{accuracy} = \frac{1}{Z^1_{accuracy}} \frac{AC}{TC} \tag{3}$$

where $Z^1_{accuracy}$ is a normalization constant.

In experiment 2, we consider a generalization of experiment 1. We first divide the range of customer selection percentages into 50 intervals with equal width (2%) and measure accuracy at the first $m$ intervals only.[5] At each interval $i \leq m$, we select the top $(2 \cdot i)\%$ of potential customers in descending order of the probability of purchasing the product and compute the ratio of the number of actual customers, $AC_i$, out of the total number of actual customers in the training data, $Tot$. We multiply the width of interval and sum those values to get the area under the lift curve over $m$ intervals. Finally we divide it by $m$ to get our final metric, $F^2_{accuracy}$. We formulate it as follows:

$$F^2_{accuracy} = \frac{1}{Z^2_{accuracy}} \frac{1}{m} \sum_{i=1}^{m} \frac{AC_i}{Tot} \cdot 2 \tag{4}$$

where $Tot = 348$, $m = 25$ and $Z^2_{accuracy}$ is an empirically derived normalization constant.

[4] If other objective values are equal, we prefer to choose a solution with small variance.

[5] This could be justified in terms of costs to handle the chosen prospects and the expected accuracy gain. As we select more prospects, the expected accuracy gain will go down. If the marginal revenue from an additional prospect is much greater than the marginal cost, however, we could sacrifice the expected accuracy gain. Information on mailing cost and customer value was not available in this study.

$F_{complexity}$: This objective is aimed at finding parsimonious solutions by minimizing the number of selected features as follows:

$$F_{complexity} = 1 - \frac{d-1}{D-1}. \quad (5)$$

Note that at least one feature must be used. Other things being equal, we expect that lower complexity will lead to easier interpretability of solutions, more consistent results, and better generalization.

## 4 Experimental Results

We test our approach in two separate experiments. In the first experiment, we try to maximize the hit rate when choosing the top of 20% potential customers as in [7]. In the second experiment, we maximize the area under the lift curve over the first 25 intervals. The best solution from experiment 2 will maximize the overall accuracy when market managers target the top 50% of prospects. We note that the best solution from experiment 1 is not necessarily the best solution in the more general case of experiment 2.

We implement a principal component analysis (PCA) followed by logistic regression (logit) for comparison purposes in both experiments. PCA's purpose is to reduce data dimensionality, analogously to our feature selection procedure, and logistic regression is used for probability estimation similarly to the ANN procedure. We also implement an intermediate model, ELSA/logit, for comparison purposes only. The ELSA/logit model differs from ELSA/ANN in the sense that the neural networks use only one hidden node.[6] We use the same criterion to select the final solution of ELSA/logit as in ELSA/ANNs. The motivation for the ELSA/logit model is to decompose the accuracy gain from the combined effects of feature selection and non-linear approximation with multiple hidden nodes. Therefore, the difference in performance between PCA/logit and ELSA/logit can be attributed to feature selection and the difference in performance between ELSA/logit and ELSA/ANNs can be attributed to non-linear approximation.

### 4.1 Experiment 1

In this experiment, we select the top 20% of customers to measure the hit rate of each solution.

We implement the PCA/logit model by first applying PCA on the training set. We select 22 PCs — the minimum required to explain more than 90% of the variance in the data set — and use them to reduce the dimensionality of the training set and the evaluation set. In order to compute the probability of purchasing an RV insurance policy for each record in the evaluation set, we use the same coefficients acquired from logistic regression on the reduced training set. We can finally compute the hit rate after selecting top 20% prospects based on the probability of purchasing a policy. In order to get the estimated hit rate of PCA/logit model, we implement 10-fold cross validation on the training set. In the cross validation procedure, the scores of the PCs are estimated using each separate training set.

We set the values for the ELSA parameters in the ELSA/ANN and ELSA/logit models as follows: $\Pr(mutation)$ = 1.0, $p_{max}$ = 1,000, $E_{cost}$ = 0.2, $\theta$ = 0.3, and $T$ = 2,000. In both models, we select the single solution which has the highest expected hit rate among those solutions with fewer than 10 features selected. We evaluated each models on the evaluation set and our results are summarized in Table 1.

| | Training set | Evaluation set | |
|---|---|---|---|
| Model (# Features) | Hit Rate ± s.d | # Correct | Hit Rate |
| PCA/logit (22) | 12.83 ± 0.498 | 109 | 13.63 |
| ELSA/logit (6) | 15.73 ± 0.203 | 115 | 14.38 |
| ELSA/ANN (7) | 15.92 ± 0.146 | 120 | 15.00 |

Table 1: Results of experiment 1. The hit rates from the three different models are shown as percentages with standard deviation. The column marked "# Correct" shows the number of actual customers who are included in the chosen top 20%. The number in parenthesis represents the number of selected features except for the PCA/logit model, where it represents the number of PCs selected.

In terms of the actual hit rate, all three models work very well. Even the lowest actual hit rate which comes from the PCA/logit model is almost 2.28 times that of random choice. ELSA/ANN returns the highest actual hit rate. In Table 1, the difference in actual hit rate between PCA/logit and ELSA/logit can be explained as the accuracy gain that comes from feature selection. In the same way, the difference in actual hit rate between ELSA/logit and ELSA/ANN can be explained as the accuracy gain that comes from non-linear approximation. Feature selection and non-linear approximation contribute about half of the total accuracy gain each.

The ELSA/ANN model results are also easier to interpret than those of the PCA/logit model. In the latter, it is not possible to determine whether each of the original features is predictive or not. It is also difficult to interpret the meaning of each of PCs in high dimensional feature spaces. The ELSA/ANN model makes it possible to evaluate the predictive importance of each features. We show the seven features that the ELSA/ANN model selects in Table 2.

| Feature Type | Selected Features |
|---|---|
| Demographic | Customer main type (average family) |
| Behavioral | Contribution to 3rd party policy, car policy, moped policy and fire policy, and number of 3rd party policies and social security policies |

Table 2: Features selected by ELSA/ANN in experiment 1.

[6] ELSA/ANN networks use $\sqrt{n_f}$ hidden nodes where $n_f$ represents the number of input features.

Among those features, we expected at least one of the car insurance related features to be selected. Further evaluation showed that prospects with at least two insured autos were the most likely RV purchasers. Moped policy ownership is justified by the fact that many people carry their mopeds or bicycles on the back of RVs. Those two features are selected again by the ELSA/logit model.[7] Using this type of information, we were able to build a potentially valuable profile of likely customers [7].

The fact that the ELSA/ANN model used only seven features for customer prediction makes it possible to save a great amount of money in data collection and database management aspects. Based on this, market managers do not need to collect or analyze other information except those seven features. This huge reduction in data dimension can reduce management cost dramatically through reduced storage requirements ($86/93 \approx 92.5\%$) and through the reduced labor and communication costs for data collection, transfer, and analysis. By contrast, the PCA/logit model needs the whole feature set to extract PCs.

We also compare the lift curves of the three models. Figure 3 shows the cumulative hit rate over the top $x\%$ of prospects ($2 \leq x \leq 100$). As expected, our ELSA/ANN model followed by ELSA/logit is the best when marketing around the top 20% of prospects. However, the performance of ELSA/ANN and ELSA/logit over all other target percentages was worse than that of PCA/logit. This is understandable because our solution is specifically designed to optimize at the top 20% of prospects while PCA/logit is not designed for specific selection points. This observation leads us to do the second experiment in order to improve the performance of ELSA/ANN model over all selection points.

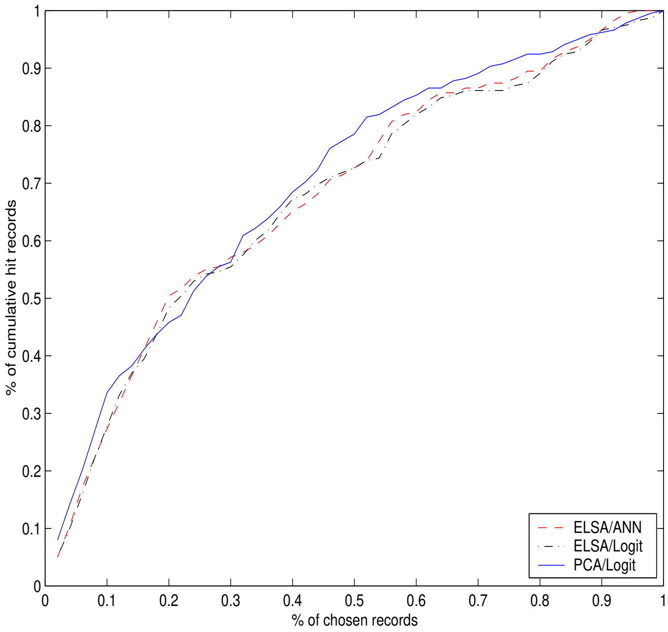


Figure 3: Lift curves of three models that maximize the hit rate when targeting the top 20% of prospects.

| | % of Selected | | | | |
|---|---|---|---|---|---|
| Model (Nr. features) | 10 | 20 | 30 | 40 | 50 |
| PCA/logit (22) | 33.6 | 45.8 | 56.3 | 68.5 | 78.6 |
| ELSA/logit (46) | 30.3 | 46.2 | 60.5 | 70.6 | 79.8 |
| ELSA/ANN (44) | 29.4 | 48.3 | 60.1 | 69.7 | 80.7 |

Table 3: Summary of experiment 2. The cumulative hit rates of the three models are shown over up to the top 50% of prospects. In practice, we optimize over the first 25 intervals which have the same width, 2%, to approximate the area under the lift curve.

### 4.2 Experiment 2

In this experiment, we search for the best solution that maximizes the overall accuracy up to the top 50% of potential customers. All three models are adjusted to maximize the overall area under the lift curve over the same intervals. In practice, we optimize over the first 25 intervals which have the same width, 2%, to approximate the area under the lift curve.

Because this new experiment is computationally much more expensive, we take a slightly different approach to choose the final solutions of ELSA/ANN and ELSA/logit. We use 2-fold cross validation estimates of all solutions and set the values of the ELSA parameters identically with the previous experiment except $p_{max}$ = 200 and $T$ = 500. Based on the accuracy estimates, we choose a solution that has the highest estimated accuracy with less than half of the original features in both models. We evaluate the three models on the evaluation set and summarize the results in Table 3 and in Figure 4.

The ELSA/ANN model works better than PCA/logit and ELSA/logit over the targeting range between 15% and 50%. In particular, ELSA/ANN is best at 15%, 20%, 25%, and 50% of targeted customers, and approximately equal to the best at 30-45%. The overall performance of ELSA/logit is better than that of PCA/logit. We attribute this to the fact that solutions from both ELSA models exclude many irrelevant features. PCA/logit, however, is competitive for targeting more than 50% of the customers, since ELSA/ANN and ELSA/logit do not optimize over these ranges.

We note that the well-established parsimony of the models selected by ELSA/ANN in experiment 1 is lost in experiment 2. We attribute this partially to the fact that different selection points may have related but different optimal subsets of features. Correlation among features seems to contribute to the lost of parsimony. For instance, a particular variable related to insurance policy ownership that is part of the optimal subset at a 20% selection rate could easily be replaced by a different, correlated feature at 30%. Even so, ELSA/ANN model is still superior to PCA/logit model in terms of the parsimony of selected features since the PCA/logit model needs

[7] The other four features selected by the ELSA/logit model are: contribution to bicycle and fire policy, and number of trailer and lorry policies.

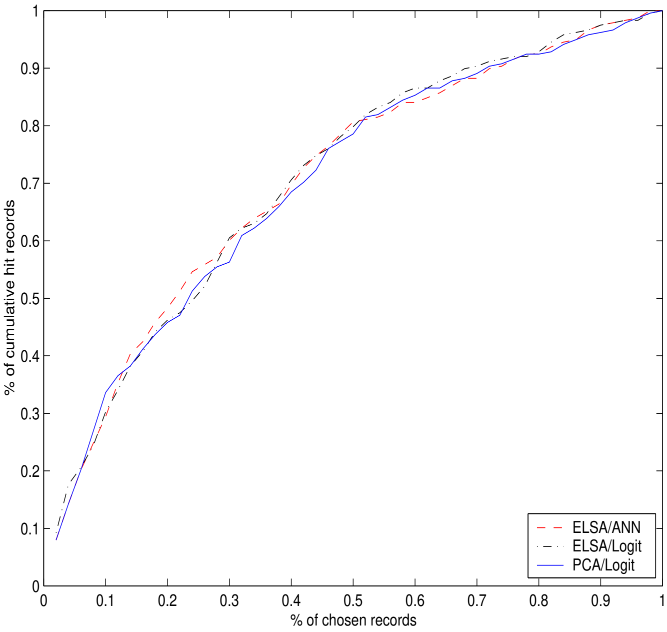


Figure 4: Lift curves of three models that maximize the area under lift curve when targeting up to top 50% of prospects.

the whole feature set to construct PCs.

## 5 Conclusion

In this paper, we presented a novel application of the multi-objective evolutionary algorithms for customer targeting. We used ELSA to search for possible combinations of features and an ANN to score customers based on the probability that they will buy the new service or product. ELSA avoids computationally expensive global comparisons among agents with a well-established environment corresponding to the set of possible values for each of the criteria. ELSA also avoids the limitations of weighted sum methods which combine multiple objectives in a subjective manner. The overall performance of ELSA/ANN in terms of accuracy was superior to the traditional method, PCA/logit, and an intermediate model, ELSA/logit. Further, the final output of the ELSA/ANN model was much easier to interpret because only a small number of features are used.

In future work we want to investigate how more general objectives affect the parsimony of selected features. We also would like to compare the performance of ELSA on the customer targeting task with other EMO algorithms. Further, we will consider a marketing campaign in a more realistic environment where various types of costs and net revenue for additional customers are considered. We could also consider budget constraints and minimum/maximum campaign sizes. This way the number of targeted customers would be determined inside an optimization routine to maximize the expected profit of the marketing campaign.

## 6 Acknowledgments

The authors wish to thank Peter van der Putten and Maarten van Someren for making the CoIL data available for this paper. This work was supported in part by NSF grant IIS-99-96044.

## Bibliography

[1] S. Bhattacharyya. Evolutionary algorithms in data mining: Multi-objective performance modeling for direct marketing. In *Proc. 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining (KDD-00)*, pages 465–473, 2000.

[2] C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, 1999.

[3] K. Deb. Evolutionary algorithms for multi-criterion optimization in engineering design. In K. Miettinen, M. Makela, P. Neittaanmaki, and J. Periaux, editors, *Proc. Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'99)*, pages 135–161, 1999.

[4] C. Emmanouilidis, A. Hunter, and J. MacIntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *2000 Congress on Evolutionary Computation (CEC-2000), San Diego, California*, pages 309–316. IEEE Service Center, July 2000.

[5] J. Horn, N. Nafpliotis, and D. Goldberg. A niched pareto generic algorithms for multiobjective optimization. In *Proc. 1st IEEE Conf. on Evolutionary Computation*, pages 82–87, Piscataway, NJ, 1994. IEEE Service Center.

[6] R. Johnson and D. Wichern. *Applied multivariate statistical analysis*. Prentice Hall, New Jersey, 3 edition, 1992.

[7] Y. Kim and W. Street. Coil challenge 2000: Choosing and explaining likely caravan insurance customers. Technical Report 2000-09, Sentient Machine Research and Leiden Institute of Advanced Computer Science, June 2000. http://www.wi.leidenuniv.nl/˜putten/library/cc2000/.

[8] Y. Kim, W. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *Proc. 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining (KDD-00)*, pages 365–369, 2000.

[9] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[10] A. Kunha, P. Oliveira, and J. Covas. Use of genetic algorithms in multicriteria optimization to solve industrial problems. In T. Back, editor, *Proc. 7th Int'l Conf. on Genetic Algorithms*, pages 682–688, San Mateo, California, July 1997. Michigan State University, Morgan Kaufmann.

[11] C. Ling and C. Li. Data mining for direct marketing: Problems and solutions. In *Proc. 4th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining (KDD-98)*, pages 73–79, 1998.

[12] F. Menczer and R. Belew. Latent energy environments. In R. Belew and M. Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Addison Wesley, Reading, MA, 1996.

[13] F. Menczer and R. Belew. Local selection. In *Evolutionary Programming VII, LNCS 1447*, Berlin, 1998. Springer.

[14] F. Menczer, M. Degeratu, and W. Street. Efficient and scalable pareto optimization by evolutionary local selection algorithms. *Evolutionary Computation*, 8(2):223–247, Summer 2000.

[15] F. Menczer, W. Street, and M. Degeratu. Evolving heterogeneous neural agents by local selection. In V. Honavar, M. Patel, and K. Balakrishnan, editors, *Advances in the Evolutionary Synthesis of Neural Systems*. MIT Press, Cambridge, MA, 2000.

[16] S. Obayashi, S. Takahashi, and Y. Takeguchi. Niching and elitist models for MOGAs. In *5th Int'l Conf. on Parallel Problem Solving from Nature (PPSN-V)*, pages 260–269, Berlin, Germany, 1998. Springer.

[17] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge, MA, 1986.

[18] J. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J. Grefenstette, editor, *Proc. Int'l Conf. on Genetic Algorithms*, pages 93–100, 1985.

[19] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[20] T. Stanley and T. Mudge. A parallel genetic algorithm for multiobjective microprocessor design. In L. Eshelman, editor, *Proc. 6th Int'l Conf. on Genetic Algorithms*, pages 597–604, San Mateo, California, July 1995. University of Pittsburgh, Morgan Kaufmann.

[21] D. Van Veldhuizen and G. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–148, 2000.